SPAM SMS FILTERING USING MACHINE LEARNING

A Mini Project Report

submitted by

MUBASHIRA A(MES20MCA-2029)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **SPAM SMS FILTERING USING MACHINE LEARNING**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bonafide work done by me under supervision of Mr. Vasudevan TV, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:Kuttippuram

Date:28-02-2022

MUBASHIRA A(MES20MCA-2029)

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **SPAM SMS FILTERING USING MACHINE LEARNING** is a bonafide record of the Mini Project work carried out by **MUBASHIRA A(MES20MCA-2029)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

At the very outset I would like to thank the almighty's mercy towards me over the years... I wish to express my sincere thanks to my project guide **Ms. Priya JD** Assistant professor, Dept.of Computer Applications who guided me for the successful completion of this project.I also thank her for valuable suggestions,guidance,constant encouragement, boundless corporation, constructive comments an motivation extended to me for completion of this project work.I would express my sincere thanks to my internal guide **Mr. Vasudevan T.V** Assistant professor, Dept.of Computer Applications for their immense guidance to complete the project successfully. I would like to express my sincere thanks to all the faculty members of Master of Computer Applications department for their support and valuable suggestion for doing the project work. Last but not least my graceful thanks to my parents, friends and also the persons who supported me directly and indirectly during the project.

MUBASHIRA A(MES20MCA-2029)

Abstract

SMS spam is a contemporary issue fundamentally because of the accessibility of very modest mass SMS bundles and the way that SMS induces higher reaction rates as it's far a depended on and personal service. In this project, we will be differentiating the messages into two categories: Ham and Spam. Ham is described as the dataset that includes the textual content of SMS messages at the side of the label indicating whether the records is legitimate message or now not. Spam is defined as the dataset that includes the textual content of SMS messages along with the label indicating the junk messages. In SMS Spam messages, the advertisers utilize the SMS text messages to target the customers with unwanted advertising. But it is troublesome, because the users pay a fee per SMS received. To overcome this, we perform a comparison between the machine learning algorithms to predict the messages and calculate the accuracy criterion by using the Spam ham dataset.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

[1]Early work proposing the application of automatic text classification techniques to SMS spam filtering includes work by Xiang, Chowdhury, and Ali (2004) who suggested that Support Vector Machines (SVMs) would be appropriate for the problem but did not evaluate their use, and work by Healy, Delany, and Zamolotskikh (2005) that considered using k-NN classifiers. Gómez Hidalgo, Bringas, Sánz, and Garcı´a (2006) evaluated a number of classification algorithms on two SMS spam datasets and concluded that these techniques can be effectively transferred from email to SMS spam filtering, with SVMs being the most suitable. Work by Cai, Tang, and Hu (2008) on a Chinese spam dataset used the simpler and lesser used Winnow algorithm (Littlestone, 1988), a linear classifier that has shown good performance in high dimensional feature domains with irrelevant features.

1.1.1 Motivation

This project motivated the issue of SMS spam mainly in relation to its potential damaging impact on consumers and mobile networks. This raises the question of what the requirements of an SMS spam solution should be. The research reviewed here is evaluated using scientific metrics such as accuracy, F-score and false positive rate. In addition to these, any complete

[1]Contents taken from the article (**Bejoy, A. and Madhu, S. Nair** (2018) Computer-aided classification of prostate cancer grade groups from MRI images using texture features and stacked sparse autoencoder, Computerized Medical Imaging and Graphics, 69, 60–68.

SMS spam solution must consider the requirements of an industrial-strength application. The motivation behind this work was to provide a lightweight client-side solution that was deployable on resource-constrained mobile devices. A potential limitation of this approach is the difficulty in the re-training required to handle concept drift.

1.2 Objective

Using cell phones and smart systems has grown more and more in recent years, and short message service has become one of the most significant communication means. Eightieth of the world's active users use mobile phone sms as a communication method. Unwanted SMS messages made for the following purposes are among some of this large variety of short messages. Sending low quality SMS and plenty of incredibly low value cell SMS kit operators. Quick message has developed into a business-trade of many billion bucks. The creation of inappropriate messages for the purpose of advertisement is harassment and the source of these messages on SMS has become the main challenge during this service. Especially SMS spam is more irritating then email spam. We are inclined to use RNN to distinguish ham and spam sequences of variable length. SVM is the most suitable method for this purpose.

1.3 Report Organization

The project report is divided into four sections. Section 2 describes literature survey. Section 3 describes the methodology used for implementing the project. Section 4 gives the results. Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

In Authors have used novel frame worked for SMS Spam Filtering make used of the two different Future selection approach based on the information gain and Chisquare metrics find out Discriminative feature representing SMS messages. The Discriminative feature subsets are then employed in two different Bayesian-based classifiers, so that SMS messages are classified as Spam or ham. Considering all experimental results, the highest overall accuracy is obtained as 90.17% by the binary classification model using top-10 CHI2 based features.

In Authors have used an algorithm is based on the Naïve Bayes and Word Occurrences table they objective to does not depend another computer filtering for Spam SMS filtering. Develop the personal and independent Spam SMS filtering to reduced the communication cost and hardware cost and also they research objective to Develop Spam filtering System on mobile phone to provide user to independent, private, secure, personal, Simple, updatable, filtering system.

In Authors used an algorithm Bayesian filtering Techniques can easily transfer to the SMS spam. Author used Combine mechanism of Naïve Bye's and dynamic nature in to single Algorithms for Spam SMS filtering. Most up Spam Detection techniques are unable to find to this Spam's because regular training of these Classifiers is not done yet, database of Spam be should be updatable Existing Spam filter are statics in nature because of that these Spam filter show false Positive and false negative Result. So Dynamic training improved the Spam filtering techniques.

In Authors have an algorithm Support vector machine (SVM) and Naïve Bayesian (NB)used for the Thai – English Spam filtering. Two method are used Spam SMS filtering the First

method Simply used Current Spam English Message Filtering filter the filtering and upgrade for Thai language support and Second method applies text normalization, word segmentation and analyzing the Thai Semantic word. Method using SVM the accuracy is high fir Thai- English but processing time is more. And method using the NB the filtering is interesting because its processing time is faster than SVM and Provide the Acceptable Accuracy.

In Authors Suggest an algorithm Gentle Boost algorithm for Spam SMS Detection because they have unbalance Data and found that Gentle Boost performed that better than other. Boosting Classifier and lead to the batter Performance the Reason for that this might be a batter method is that it works batter for unbalanced and binary classification.

In Authors have used an algorithm is based on Bayesian filtering and Spam filtering technology is based on the black list, white list and keyword has certain defect, at present filtering of Spam SMS turns to Study based on the Content Filtering.

Chapter 3

Methodology

3.1 Introduction

In this project, One of the most used techniques as the base classifier to overcome the spam problem is the Support Vector Machine. SVM Classify spam that is to differentiate between spam and authorized messages. First Dataset containing 5000 messages samples containing both spam and ham(non-spam) type messages is used.Where 60% messages are used for training and remaining 40% are used for testing. Later vocabulary is built, which contains a set of most frequently words chosen from the training/testing set. Vocabulary is then used to calculate the TF-IDF(term frequency-inverse document frequency) values, where each messages will be represented on the basis of the importance of word in the entire dataset which is a N dimensional vector. This vector is feature vector. A Machine learning algorithm, Support Vector Machine(SVM) is trained to classify the given messages. Each of these messages belongs to only one of two classes. The idea of SVM classification is find a linear separation boundary that correctly classifies training samples. And later this model is used to predict new messages given, which is the main aim or the system..

3.2 Algorithm-SVM

Support Vector Machine, abbreviated as SVM, is used for every task of regression and classification. But, it's commonly used in goals for classification. The aim of the support vector machine algorithm is to search for a hyperplane in the associated N-dimensional space (N —

the quantity of features) which separately classifies the information points. There are several potential hyperplanes which would be chosen to separate the 2 categories of information points. Our goal is to search for a plane with the utmost margin, that is, the utmost distance between the points of knowledge of each category. Increasing the margin gap should provide some reinforcement so that future data points can be identified with extra confidence.

3.3 Workflow

Data Collection: In this phase authers have collected a dataset based on which they have performed the experimentation from Kaggle Repository.

Data cleaning: in this phase the authors have cleansed all the data which were taken into consideration. Authors have removed all the white-spaces, lowered the alphabet so that works like Equal and equal become the same , remove the remaining punctuation, like! is not that much important, tokenize each message,to represent the message as a list of words and done stemming, converting all the words to their root word, like floor ,floored to floor.

Generating Testing and Training Data Sets : Authors have created the testing and training data on the converted cleansed datasets.

Generating Word Cloud Vector : Authors have used the TF-IDF vectorization for creating the word vector.On the basis, the spam feature will be classified.

Prediction : Authors have given input messages to check whether the message is Spam or Ham.

This project was developed using Agile Development Model. The entire project was divided into four sprints. In the first sprint, collect the dataset. The designing of front-end and development of back-end was done in the second.Third sprint training and testing and the fourth sprint output generated.

3.4 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task duration, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it. Project plan is given below figure. The project has four sprints.

User Story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	30/11/21	30/11/21	2	Completed
2		15/12/21	15/12/21		Completed
3	Sprint 2	29/12/21	30/12/21	4	Completed
4		15/01/22	16/01/22		Completed
5	Sprint 3	22/01/22	23/01/22	4	Completed
6		29/01/22	30/01/22		Completed
7	Sprint 4	05/02/22	06/02/22	3	Completed
8		12/02/22	12/02/22		Completed

Figure 3.1: Project Plan

3.5 User Story

A key component of agile software development is putting people first, and user-stories put actual end users at the center of the conversation. Stories use non-technical language to provide context for the development team and their efforts. After reading a user story, the team knows why they are building what they're building and what value it creates. A user story is a tool used in agile software development to capture a description of a software feature from an end user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work which drives collaboration, creativity, and a better product overall. The user story of system is

given in the below figure.

User Story ID	As a type of User	I want to <perform some task>	So that I can < Achieve Some Goal>
1	Admin	Login	I can get access to the system
2	User	Register	Registering users
3	User	Login	I can get access to the system
4	User	Result data collection form	Printing form
5	User	Training the data	View accuracy of the model
6	User	Modelling the data	Model created with SVM Algorithm
7	User	Testing the data	Input new data
8	User	Output generation	Output generated

Figure 3.2: User Story

3.6 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The product backlog is the single authoritative source for things that a team works on. That means that nothing gets done that isn't on the product backlog. Conversely, the presence of a product backlog item on a product backlog does not guarantee that it will be delivered. It represents an option the team has for delivering a specific outcome rather than a commitment.It should be cheap and fast to add a product backlog item to the product backlog, and it should be equally as easy to remove a product backlog item that does not result in direct progress to achieving the desired outcome or enable progress toward the outcome. The Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user-centric e.g. in the form of user stories.The product backlog of the system is given below figure.

User Story ID	Priority < High /Medium /Low >	Size (Hours)	Sprint <#>	Status < Planned / InProgress / Completed >	Release Date	Release Goal
1	High	5	1	Completed	30/11/21	Collect spam ham dataset from Kaggle website
2	Medium	5		Completed	15/12/21	Data pre-processing
3	Medium	5	2	Completed	29/12/21 -30/12/21	User Interface Design
4	High	10		Completed	15/01/22 -16/01/22	Result data collection form
5	High	7	3	Completed	22/01/22 - 23/01/22	Training the dataset
6	High	8		Completed	29/01/22 - 30/02/22	Apply SVM Algorithm and create model
7	High	8	4	Completed	05/02/22 - 06/02/22	Testing and predicting (whether it is spam or not)
8	High	2		Completed	12/02/22	Output generation

Figure 3.3: Product Backlog

3.7 Sprint Backlog Plan

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story. Most teams also estimate how many hours each task will take someone on the team to complete.

Backlog Item	Status and Completion date	Original Estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13
User story #1,2			Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours
Dataset Collection	30/11/21	5	5	0	0	0	0	0	0	0	0	0	0	0	0
Analysis	15/12/21	5	0	5	0	0	0	0	0	0	0	0	0	0	0
User story #3,4															
UI Designing	30/12/21	5	0	0	3	2	0	0	0	0	0	0	0	0	0
Coding	16/01/22	10	0	0	0	0	5	5	0	0	0	0	0	0	0
User story #5,6															
Training	30/01/22	15	0	0	0	0	0	0	3	4	4	4	0	0	0
User story #7,8															
Testing	12/02/22	10	0	0	0	0	0	0	0	0	0	0	4	4	2
Total		50	5	5	3	2	5	5	3	4	4	4	4	4	2

Figure 3.4: Sprint Backlog Plan

3.8 Sprint Backlog Actuals

Actual sprint backlog is what adequate sprint planning is actually done by project team there may or may not be difference in planned sprint backlog. The detailed sprint backlog (Actual) is given below

Backlog Item	Status and Completion date	Original Estimate in hours	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Completed <Y/N>
User story #1,2			Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	Hours	
Dataset Collection	30/11/21	5	5	0	0	0	0	0	0	0	0	0	0	0	0	Y
Analysis	15/12/21	5	0	5	0	0	0	0	0	0	0	0	0	0	0	Y
User story #3,4																
UI Designing	30/12/21	5	0	0	3	2	0	0	0	0	0	0	0	0	0	Y
Coding	16/01/22	10	0	0	0	0	5	5	0	0	0	0	0	0	0	Y
User story #5,6																
Training	30/01/22	15	0	0	0	0	0	0	3	4	4	4	0	0	0	Y
User story #7,8																
Testing	12/02/22	10	0	0	0	0	0	0	0	0	0	0	4	4	2	Y
Total		50	5	5	3	2	5	5	3	4	4	4	4	4	2	

Figure 3.5: Sprint Backlog Actuals

Chapter 4

Results and Discussions

4.1 Datasets

The dataset is from Kaggle, a collection of spam SMS messages, with 5572 messages, all classified as either ‘ham’ or ‘spam’ . The dataset contains 13.4% spam and 86.6% ham.

4.2 Results

The cause of this Project is to discover the effects of making use of gadget gaining knowledge of techniques to discover Message spam detection. SMS unsolicited mail (every now and then known as cell smartphone junk mail) is any junk message brought to a cellular phone as textual content messaging via the Short Message Service (SMS). The dataset for this mission originates from the UCI Machine Learning Repository.Using special techniques to establish relation between the textual content and the category SPAM or HAM like, primarily based on length of message, word depend, unique keywords. Then construct class fashions the use of one-of-a-kind strategies to differentiate spam SMS.Spam Ham dataset, SVM algorithm gives the better accuracy whether the message contains HAM or SPAM.Total Dataset: 5572

4.3 Screenshots

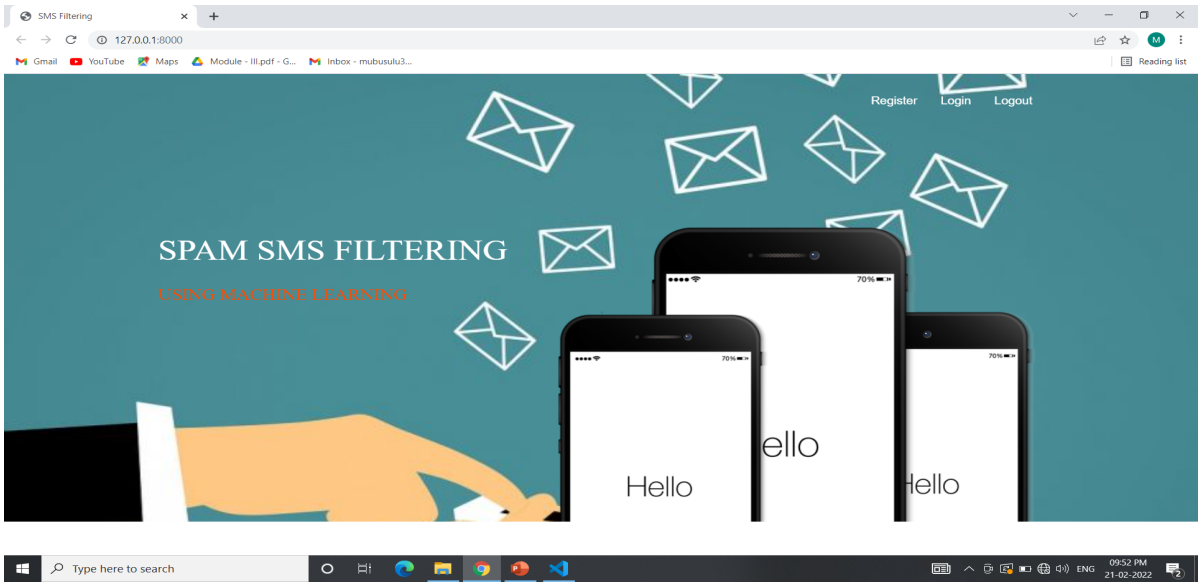

Figure 4.1: User Interface 1

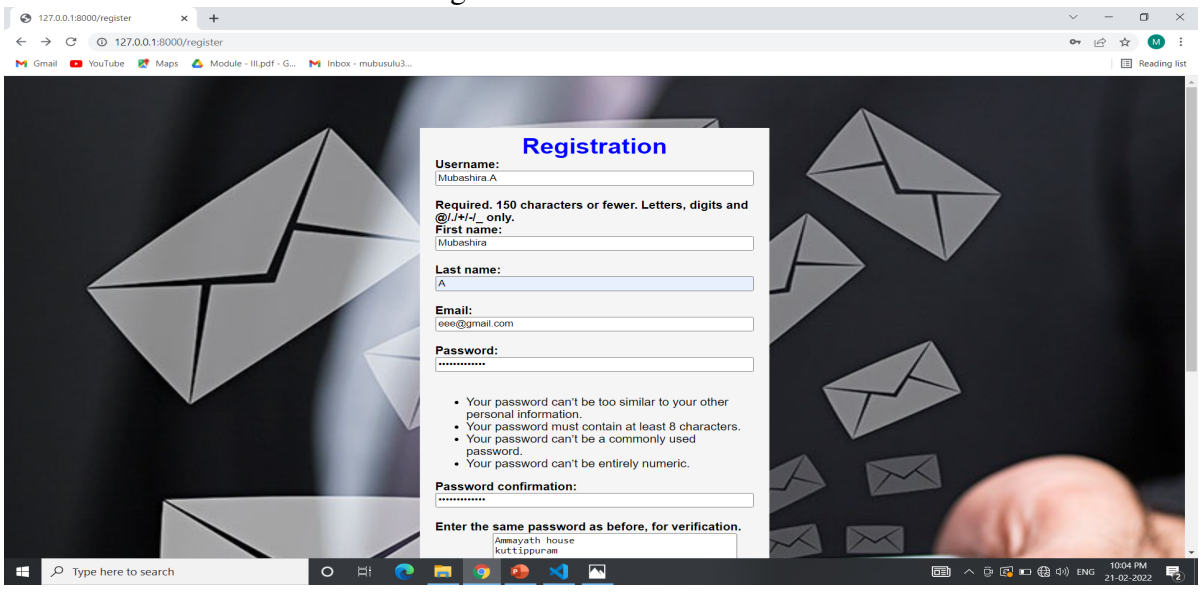

Figure 4.2: User Interface 2

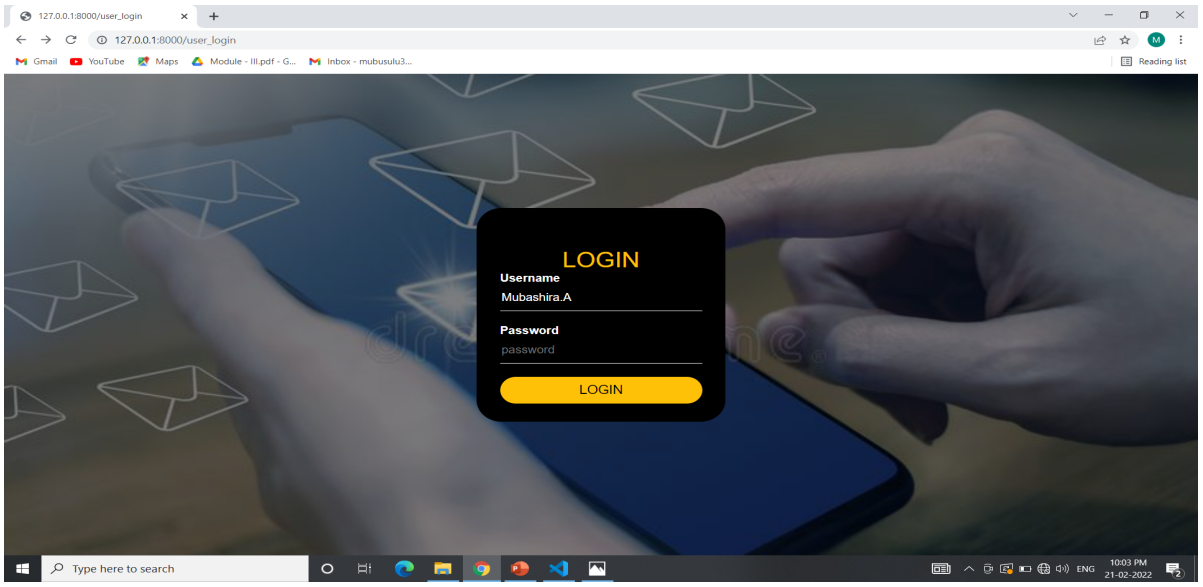

Figure 4.3: User Interface 3

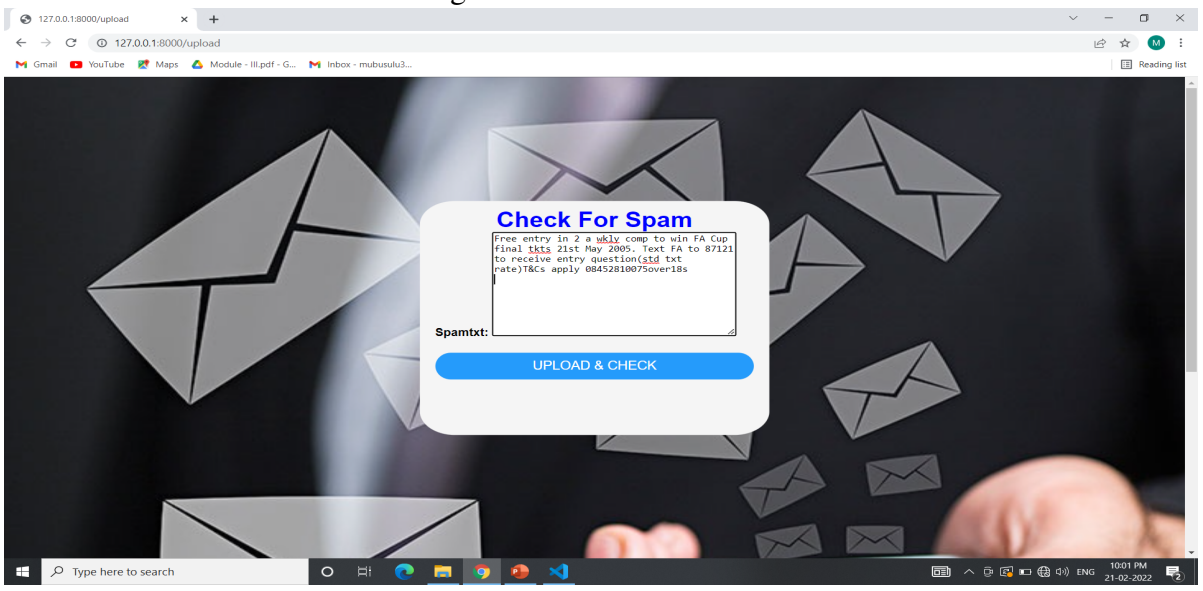

Figure 4.4: User Interface 4

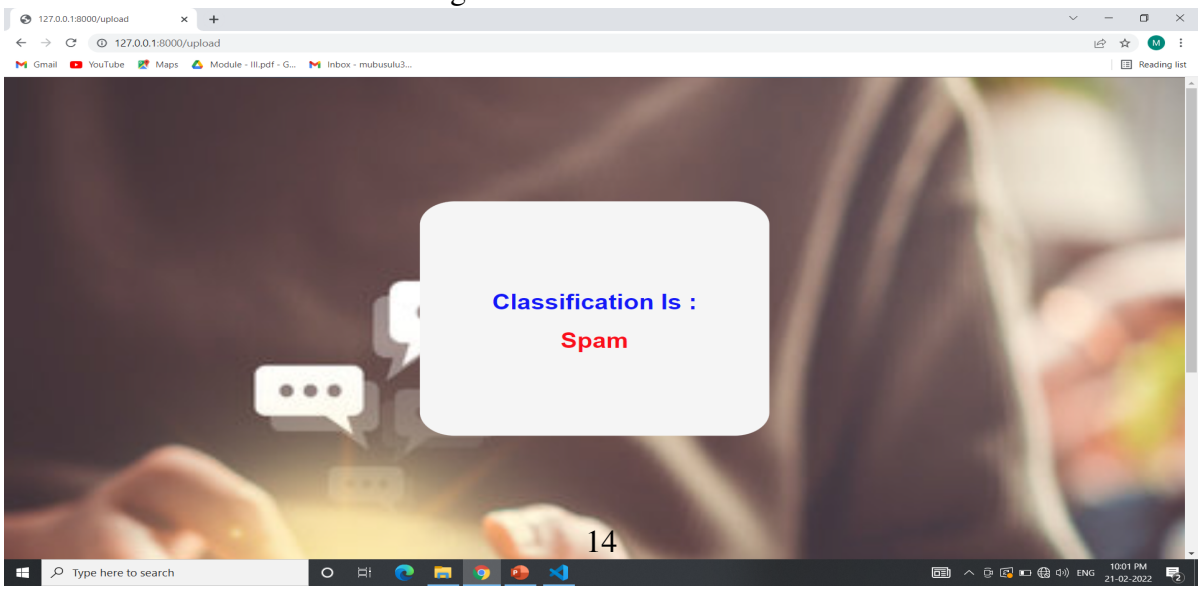

Figure 4.5: User Interface 5

Chapter 5

Conclusions

The undertaking of automated filtering of SMS messages is still a trouble. There are two predominant challenges hindering the improvement of algorithms: lack of actual datasets, and occasional functions that extracts the messages. The main aim of this project is to SVM algorithm with better accuracy score. By using kernels in this, we can find the better output through SVM with linear kernel. The text messages are differentiated with Ham or Spam. It predicts whether the message in the dataset is Ham or Spam and predicts the performance through accuracy criterion. Using this algorithm, we can see whether the data in the dataset is predictable or not.

References

[1] **S. M. Abdulhamid et al** ., "A Review on Mobile SMS Spam Filtering Techniques," in IEEE Access, vol.5.

[2] **JOUR Lota, Lutfun, Hossain** (1992) A simple weight decay can improve generalization, Advances in Neural Information Processing Systems, 950-957.

[3] **CHAP, Choudhary, Neelam, Jain, Ankit**, "Towards Filtering of SMS Spam Messages Using Machine Learning Based Techniques".

[4] **Nagwani, N. K., Sharaff** , A, "SMS spam filtering and thread identification using bi-level text classification and clustering techniques", Journal of Information Science,2017.

[5] "SMS Spam Detection using Machine Learning Approach", Houshmand Shirani-mehr,2013.

[6] **Yadav, Kuldeep, Saha, Swetank, Kumaraguru, Ponnurangam, Kumra, Rohit, "Take control of your SMSes:** : Designing an usable spam SMS filtering system", IEEE 13th International Conference on Mobile Data Management, MDM 2012

Appendix

Source Code

```
import pandas as pd
import seaborn as sns
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.http import HttpResponse, HttpResponseRedirect
from django.shortcuts import redirect, render
from django.urls import reverse
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
df1 = pd.read_csv("D:\\MCA\sem 3\\Mini project\\spamham.csv")
df = df1.where((pd.notnull(df1)), '')
df.loc[df["Category"] == 'ham', "Category",] = 1
df.loc[df["Category"] == 'spam', "Category",] = 0
# split data as label and text . System should be capable of predicting the label based on the text
df_x = df['Message']
df_y = df['Category']
from .forms import *
from .models import *
# split the table - 80 percent for training and 20 percent for test size
x_train, x_test, y_train, y_test = train_test_split(df_x, df_y, train_size=0.6, test_size=0.4, random_state=4)
# feature extraction, coversion to lower case and removal of stop words using TFIDF VECTORIZER
tfvec = TfidfVectorizer(min_df=1, stop_words='english', lowercase=True)
x_trainFeat = tfvec.fit_transform(x_train)
x_testFeat = tfvec.transform(x_test)
# SVM is used to model
y_trainSvm = y_train.astype('int')
classifierModel = LinearSVC()
classifierModel.fit(x_trainFeat, y_trainSvm)
predResult = classifierModel.predict(x_testFeat)

def index(request):
   return render(request,'index.html')
def upload2(request):
   return render(request,'upload2.html')
def register(request):
   reg=False
   if request.method=="POST":
      user_form=UserForm(data=request.POST)
      profile_form=ProfileForm(data=request.POST)
      if user_form.is_valid() and profile_form.is_valid():
         user=user_form.save()
         user.save()
```

Appendix

```
            profile=profile_form.save(commit=False)
            profile.user=user
            profile.save()

            reg=True
        else:
            HttpResponse("Invalid Form")
    else:
        user_form=UserForm()
        profile_form=ProfileForm()
    return render(request,'register.html',{'register':reg,'user_form':user_form,'profile_form':profile_form})
def upload(request):
    reg=False
    res=''
    if request.method=="POST":
        spam=request.POST.get('spamtxt')
        print(spam)
        uploadform=UploadForm(data=request.POST)
        message_to_test=[spam]
        x_testFeat = tfvec.transform(message_to_test)
        predResult = classifierModel.predict(x_testFeat)
        print(predResult)
        if(predResult[0]==1):
            res='No Spam'
        else:
            res='Spam'
        print('Classification is',res)
        if uploadform.is_valid():
            #user=uploadform.save()
            #user.save()
            reg=True
            return render(request,'upload2.html',context={'result':res})
        else:
            HttpResponse("Invalid Form")
    else:
        uploadform=UploadForm()
    return render(request,'upload.html',context={'result':res,'uploadform':uploadform})
def user_login(request):
    if request.method=="POST":
        username=request.POST.get('username')
        password=request.POST.get('password')

        user=authenticate(username=username,password=password)
        if user:
            if user.is_active:
                login(request,user)
                return HttpResponseRedirect(reverse('dashboard'))
            else:
                return HttpResponse('Not active')
        else:
            return HttpResponse("Invalid username or Password")
    else:
        return render(request,'login.html')
def dashboard(request):
    return render(request,'dashboard.html')
@login_required
def user_logout(request):
    logout(request)
    return redirect('index')
```